

Automatyzacja definiowania kształtu noska rakiety przeciwlotniczej w programie AutoCAD

Process automatization of creating shape of anti-aircraft missile nose in AutoCAD software

JACEK WARCHULSKI
MARCIN WARCHULSKI*

DOI: <https://doi.org/10.17814/mechanik.2019.7.54>

W artykule przedstawiono potencjalne zastosowanie systemów CAx w procesie automatyzacji zadań graficznych. Przeanalizowano możliwość wykorzystania języka Delphi i technologii ActiveX do definiowania kształtu noska rakiety przeciwlotniczej w systemie CAx.

SŁOWA KLUCZOWE: programowanie, Delphi, ActiveX, AutoCAD, obiekty 3D

In this paper possibilities of using of CAx systems to automation process of graphical tasks are presented. Possibility of using Delphi language and ActiveX technology to creating the shape of an anti-aircraft missile in CAx system are analysed.
KEYWORDS: programming, Delphi, ActiveX, AutoCAD, 3D objects

Wprowadzenie

Właściwe wykorzystanie dużej ilości energii silnika raketowego, przekształconej w stosunkowo krótkim czasie, wymaga zapewnienia jak najkorzystniejszych warunków ruchu. Konstruktorzy uzbrojenia raketowego stosują więc metody optymalizacji w możliwie najszerszym stopniu. Główną rolę w poszukiwaniu optymalnych warunków lotu odgrywają dwa zagadnienia. Po pierwsze należy dobrać optymalne programy sterowania, w tym programy pracy zespołów napędowych oraz programy sterowania kierunkiem lotu rakiet, a po drugie – kształty rakiet o najmniejszym oporze aerodynamicznym [6].

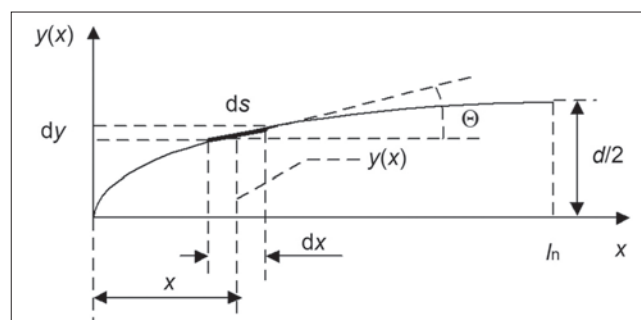
Do zasadniczych kryteriów w dynamice lotu rakiety należą minimalny czas osiągnięcia celu oraz minimalna energia potrzebna na przebycie drogi do celu.

Gdy rozpatruje się ruch pocisku w atmosferze, kryterium minimalnoczasowe nie uwzględnia strat energetycznych, które są tym większe, im większa jest prędkość lotu rakiety. Zwiększenie prędkości powinno skutkować skróceniem czasu lotu. Nie oznacza to jednak, że w przypadku rakiet sterowanych programem minimalnoczasowym straty energii nie odgrywają żadnej roli. Jedynym sposobem ich ograniczenia jest jednak zmniejszenie oporu aerodynamicznego. Właśnie dlatego poszukiwanie kształtu ciała o najmniejszym oporze aerodynamicznym jest jednym z najważniejszych problemów.

Najkorzystniejszego obrysu kadłuba rakiety można poszukiwać dla różnych zakresów prędkości lotu. Różnym zakresom prędkości odpowiadają różne hipotezy odbicia powietrza. Poszukiwanie optymalnego obrysu noska da się sprowadzić do minimalizacji tylko oporu ciśnieniowe-

go bez uwzględniania tarcia. Opór tarcia można natomiast uwzględnić dopiero przy ustalaniu najkorzystniejszego wydłużenia noska. Takie podejście jest uzasadnione, ponieważ siły tarcia jako styczne do obrysu wpływają na jego kształt w nieznacznym stopniu, lecz są istotne przy ustalaniu smukłości noska.

Funkcję opisującą charakter krzywej kształtu noska rakiety przeciwlotniczej można określić za pomocą rachunku wariacyjnego. Zadanie sprowadza się do wyznaczenia oporu obrysu rozszerzającej się części kadłuba definiowanego jako ciało obrotowe (rys. 1).



Rys. 1. Elementarny rozkład siły oporu (d – średnica noska, ds – elementarna droga, x – zmienna niezależna)

Funkcja opisująca kształt noska o minimalnym oporze aerodynamicznym jest zdefiniowana następująco:

$$f(x) = \frac{d}{2} \left(\frac{x}{l_n} \right)^{\left(\frac{n}{n+1} \right)}$$

gdzie: x – zmienna niezależna, d – średnica noska, l_n – długość noska.

Dla tak zdefiniowanej funkcji oraz dla kształtu części cylindrycznej kadłuba (rys. 2) całkowity współczynnik oporu rakiety przeciwlotniczej jest sumą współczynnika oporu ciśnieniowego C_{xc} i oporu tarcia C_{xt} [6]:

$$C_x = C_{xc} + C_{xt} = \frac{16}{2^{(n+1)}} \cdot \frac{1}{\lambda_n^{(n-1)}} \left(\frac{n}{n+1} \right)^n \cdot k + 4C_t \left(\frac{n+1}{2n+1} \cdot \lambda_n + \lambda_c \right)$$

gdzie: λ_n – wydłużenie noska, λ_c – wydłużenie części cylindrycznej.

* Mjr dr inż. Jacek Warchulski, jacek.warchulski@wat.edu.pl, <https://orcid.org/0000-0003-4451-5711> – Wydział Mechatroniki i Lotnictwa, Wojskowa Akademia Techniczna, Warszawa, Polska
Mjr dr inż. Marcin Warchulski, marcin.warchulski@wat.edu.pl, <https://orcid.org/0000-0003-1310-3001> – Wydział Mechatroniki i Lotnictwa, Wojskowa Akademia Techniczna, Warszawa, Polska



Rys. 2. Optymalizacja kształtu rakiety przeciwlotniczej dla opływu nadźwiękowego ($n=3$ oraz $k = \frac{4}{\sqrt{M^2-1}}$, C_f – współczynnik tarcia płaskiej płytki zależny od liczby Reynoldsa)

Dostęp do obiektów programu AutoCAD

Systemy modelowania swobodnego nie zawierają funkcji umożliwiających automatyczne generowanie współrzędnych punktów węzłowych polilinii czy regionów, które są z kolei wykorzystywane do tworzenia obiektów brylowych. Z tych względów zdecydowano się na opracowanie aplikacji, która pozwoliłaby na generowanie geometrii modelu przestrzennego noska rakiety przeciwlotniczej z zastosowaniem programu AutoCAD i darmowego środowiska programistycznego.

Programowanie zadań grafiki jest możliwe z wykorzystaniem interfejsu automatyzacji oraz języków programowania wbudowanych w systemy CAx [1–3]. Wymagania postawione w procesie generowania modelu rakiety przeciwlotniczej spełnia interfejs typu klient–serwer, przy czym rolę klienta automatyzacji pełni program napisany w języku Delphi z wykorzystaniem środowiska programistycz-

nego Embarcadero RAD Studio, natomiast rolę serwera automatyzacji – program AutoCAD 2018.

Automatyzacja OLE obejmuje możliwość programowego sterowania obiektami innego programu oraz protokół, za pomocą którego dany program może uzyskać dostęp do obiektu znajdującego się w innym programie. Nawiązanie połączenia z serwerem OLE programu AutoCAD jest możliwe dzięki funkcji *CreateOleObject*, która po wywołaniu tworzy obiekt udostępniający obiekty programu AutoCAD. Natomiast do uzyskania dostępu do aktualnie otwartego rysunku wykorzystuje się funkcję *GetActiveOLEObject*, udostępniającą obiekty aktywnego programu zewnętrznego. Połączenie z serwerem OLE dla programów tworzonych w języku Delphi zapewnia moduł *PolaczAutoCAD* (rys. 3). Znajdująca się w tym module procedura *PolaczZACAD* pozwala uzyskać dostęp do przestrzeni modelu uruchomionego systemu CAx.

```

unit PolaczAutoCAD; {nazwa modułu}
interface {deklaracje widoczne dla innych programów i modułów}
uses {lista modułów zawierających importowane elementy}
  ComObj, Variants, Dialogs; {moduł obsługi obiektów OLE i Dialogs}
var {deklaracje zmiennych}
  AcadApp, AcadDoc, AcadDwg :OleVariant;
  {pod zmienną AcadApp będzie przypisany obiekt Autocad.Application}
  {pod zmienną AcadDoc będzie przypisany obiekt Autocad.Application.Document}
  {a pod zmienną AcadDwg obiekt Autocad.Application.Document.ModelSpace}
function PolaczZACAD :Boolean; {deklaracja funkcji}
implementation {deklaracje niewidoczne dla innych programów i modułów}
function PolaczZACAD :Boolean;
begin
  Result:=False; {przypisanie wartości funkcji PolaczZACAD}
  try {początek zewnętrznego bloku chronionego}
    try {początek wewnętrznego bloku chronionego}
      {jeśli AutoCAD jest uruchomiony}
      AcadApp := GetActiveOLEObject('AutoCAD.Application');
    Except {jeśli AutoCAD jest uruchomiony to należy stworzyć obiekt}
      AcadApp := CreateOleObject('AutoCAD.Application');
      AcadApp.Visible:=True;
    end; {koniec wewnętrznego bloku chronionego}
    AcadDoc:=AcadApp.ActiveDocument;
    if AcadApp.Documents.Count > 0 then
    Begin {rysunek przypisujemy go do zmiennej AcadDwg}
      AcadDwg:=AcadDoc.ModelSpace;
      Result:=True; {jeśli uzyskano dostęp do obiektu przyjmujemy wartość True}
    end
    else {jeśli nie jest otworzony to wyświetlony zostanie komunikat}
      ShowMessage('Najpierw proszę założyć lub otworzyć jakiś dokument...');
    except {w razie niepowodzenia w dostępie od obiektu AutoCAD}
      ShowMessage('Brak aktywnych rysunków lub aktywne polecenie.');
```

Rys. 3. Kod modułu *PolaczAutoCAD* [4]

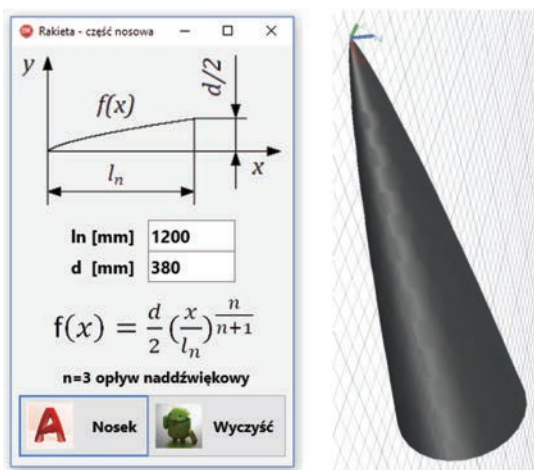
TABLICA. Wybrane metody interfejsu automatyzacji OLE

Składnia	Zastosowanie
RetVal := object.AddLine(StartPoint, EndPoint) *RetVal – Return a Value	Rysuje odcinek między dwoma punktami
RetVal := object.AddLightweightPolyline (VerticesList) * VerticesList – macierz wierzchołków (array of Doubles – wielokrotność liczby dwa)	Rysuje polilinię 2D na podstawie podanej listy wierzchołków
RetVal := object.AddRegion (ObjectList) * ObjectList – tablica obiektów tworzących region (łuki, okręgi, linie, elipsy, polilinie lub splajny)	Tworzy region ze zbioru składników (obiekty muszą tworzyć zamknięty obszar)
RetVal := object.AddRevolvedSolid (Profile, AxisPoint, AxisDir, Angle) * Profile – obiekt typu region * AxisPoint – współrzędne punktu początkowego osi obrotu * AxisDir – wektor 3D określający kierunek osi obrotu * TaperAngle – kąt w radianach	Rysuje bryłę obrotową na podstawie danego profilu i osi obrotu

Przykład automatyzacji wprowadzania obiektów przestrzennych

Środowisko programistyczne RAD Studio 10.2 Tokyo Starter oraz język programowania Delphi posłużyły do opracowania programu komputerowego *Nosek.exe* [4], umożliwiającego wprowadzenie w obszarze modelu programu AutoCAD jednolitej bryły noska rakiety przeciwlotniczej.

```
var Nosek : OLEVariant;
{ ... }
Polilinia_Zarys[ind_poli]:=
AcadDwg.AddLine(p2D(x1,y1), p2D(x1,0));
Polilinia_Zarys[ind_poli+1]:=
AcadDwg.AddLine(p2D(x1,0), p2D(0,0));
//utworzenie regionu i bryły obrotowej
Region1:=AcadDwg.AddRegion(Polilinia_Zarys);
Nosek:=AcadDwg.AddRevolvedSolid
(Region1[0], p2D(0,0), p2D(x1,0), 2*Pi);
//Usuwanie obwiedni
for i:=0 to ind_poli+1 do
Polilinia_Zarys[i].Delete;
Region1[0].Delete;
//Zoom
AcadDoc.Application.ZoomExtents;
{ ... }
```

Rys. 4. Fragment kodu programu *Nosek.pas* – utworzenie bryły [4]Rys. 5. Działanie programu *Nosek.exe* w programie AutoCAD [4]

Bryłę noska rakiety można stworzyć za pomocą standardowych poleceń rysunku przestrzennego, bazując na profilach definiujących daną bryłę (można wykorzystać zarówno obiekty typu polilinia, jak i obiekty typu region).

Aby zautomatyzować ten proces, trzeba zdefiniować wybrane metody interfejsu automatyzacji OLE programu AutoCAD 2018 (patrz tablica), dotyczące wprowadzania obiektów płaskich i przestrzennych [5].

Funkcja opisująca kształt noska posłużyła do utworzenia profilu zamkniętego typu region, na podstawie którego utworzono obiekt bryłowy z wykorzystaniem polecenia *AddRevolvedSolid*. Fragment kodu głównego modułu *Nosek.pas*, umożliwiającego zdefiniowanie bryły poprzez przekręcenie regionu oraz usunięcie obwiedni przekręcenia, przedstawiono na rys. 4.

Działanie programu *Nosek.exe* pokazano na rys. 5.

Podsumowanie

Dzięki programom komputerowego wspomagania projektowania, użytkownicy mogą zautomatyzować powtarzalne zadania projektowe. Do tego celu można wykorzystać specjalne, nowo opracowane programy, które nie są standardowo dołączane do programów typu CAD [7–11].

Automatyczne definiowanie składników rysunku programu AutoCAD może mieć szerokie praktyczne zastosowanie. Zaprezentowany w artykule przykład użycia języka Delphi potwierdza, że wykorzystanie języków programowania pozwala na automatyzację modelowania obiektów płaskich i przestrzennych w systemach komputerowego wspomagania, co pośrednio prowadzi również do bardziej produktywnego wykorzystania danych systemów CAX.

LITERATURA

- [1] Dudek M. „AutoLISP. Praktyczny kurs”. Helion, 1997.
- [2] Czyżyczyk W., Lisowski E. „Automatyzacja zadań grafiki za pomocą Delphi”. Helion, 2002.
- [3] Warchulski J., Warchulski M. „Przykłady automatyzacji zadań grafiki w programie AutoCAD”. XIV Międzynarodowa Szkoła Komputerowego Wspomagania Projektowania, Wytwarzania i Eksploatacji, Jurata. (10–14 maja 2010): 515–522.
- [4] Warchulski J., Warchulski M. „Program Nosek.exe”. Warszawa: Wojskowa Akademia Techniczna, 2019.
- [5] <https://knowledge.autodesk.com/search-result/caas/CloudHelp/cloudhelp/2016/ENU/AutoCAD-ActiveX/files/GUID-5D302758-ED3F-4062-A254-FB57BAB01C44-htm.html>.
- [6] Dubiel S. „Uogólniona postać rozwiązania obrysu noska osiowo-symetrycznego o najmniejszym oporze aerodynamicznym”. *Mechanika Teoretyczna i Stosowana*. 28, 3–4 (1990): 513–524.
- [7] Warchulski J., Warchulski M. „Wykorzystanie programu AutoCAD do generowania punktów węzłowych polilini”. *Mechanik*. 7 (2014): 713–720.
- [8] Warchulski J., Warchulski M. „Wykorzystanie programu AutoCAD do generowania kształtu rakiety przeciwlotniczej”. *Mechanik*. 7 (2015): 931–942. DOI: <http://dx.doi.org/10.17814/mechanik.2015.7.314>.
- [9] Warchulski J., Warchulski M. „Automatyzacja użycia bloków z atrybutami w programie AutoCAD”. *Mechanik*. 7 (2016): 862–863. DOI: <http://dx.doi.org/10.17814/mechanik.2016.7.209>.
- [10] Warchulski J., Warchulski M. „Automatyzacja definiowania kształtu profili lotniczych w programie AutoCAD”. *Mechanik*. 7 (2017): 609–611. DOI: <https://doi.org/10.17814/mechanik.2017.7.88>.
- [11] Warchulski J., Warchulski M. „Automatyzacja definiowania obiektów przestrzennych w programie AutoCAD”. *Mechanik*. 7 (2018): 564–566. DOI: <https://doi.org/10.17814/mechanik.2018.7.85>. ■